FCN-biLSTM Based VAT Invoice Recognition and Processing

Fei Jiang[1,2(✉)], Huan Chen[1,2], and Liang-Jie Zhang[1,2]

[1] National Engineering Research Center for Supporting Software of Enterprise Internet Services, Shenzhen, China
[2] Kingdee Research, Kingdee International Software Group Company Limited, Shenzhen, China
faye_jiang@kingdee.com

Abstract. Financial Sharing Centre of big or medium-sized enterprises that need to handle a large number of VAT invoices every day, but these invoices are often handled manually in poor efficiency. They need automation of unsupervised processing systems for VAT invoices to reduce costs and also to promote their financial management capability. In this paper, we develop FCN-biLSTMs that are capable of processing and recognizing invoice automatically. In view of the characteristics of invoice, we propose the methods that extract text lines by using invoice layout information and text characteristics, and achieve higher accuracy. Combined with the previous text detection methods and the attention-based biLSTM sequence learning structure for text recognizing, we developed an automatic VAT invoice recognition and processing system. The system in the actual projects of enterprises has achieved impressive performance.

Keywords: FCN · biLSTM · Invoice recognition and processing

1 Introduction

Chinese domestic value-added tax (VAT) invoice is an important accounting and billing document and is a corporate tax certificate, and it is widely present in dealings among enterprises. The format of it is under strict control of State Administration of Taxation. Financial Sharing Centre of big or medium-sized enterprises need to handle a large number of VAT invoices every day, but these invoices are often handled manually in poor efficiency. They need automation of unsupervised processing systems for VAT invoices to reduce costs and also to promote their financial management capability [1]. There are some projects of this kind that have been built or have been bringing forth to build. The undergoing of an enterprise internal ERP plans is providing a good infrastructure for it, and also, the developing of image processing technologies such as text detection, text recognition and others are coming into a state of commercial feasibility for it, with some extra efforts we can turn the VAT invoice image recognition and processing automation into reality.

Due to the large variability of text patterns and the highly complicated background, the recognition and processing for photo VAT invoice images are much more challenging than the scanned ones. An overview of the network architecture is presented in

S. Liu et al. (Eds.): EDGE 2018, LNCS 10973, pp. 135–143, 2018.
https://doi.org/10.1007/978-3-319-94340-4_11

Fig. 1. It consists of a number of convolutional layers, corner points of text bounding boxes, segmentation maps for text, and layout information for regressing the text box locations, encoder for embedding proposals of varying sizes to fixed-length vectors, and an attention-based Long Short-Term Memory (LSTM) decoder for word recognition. Via this framework, an automatic VAT invoice recognition and processing system is built and implemented.

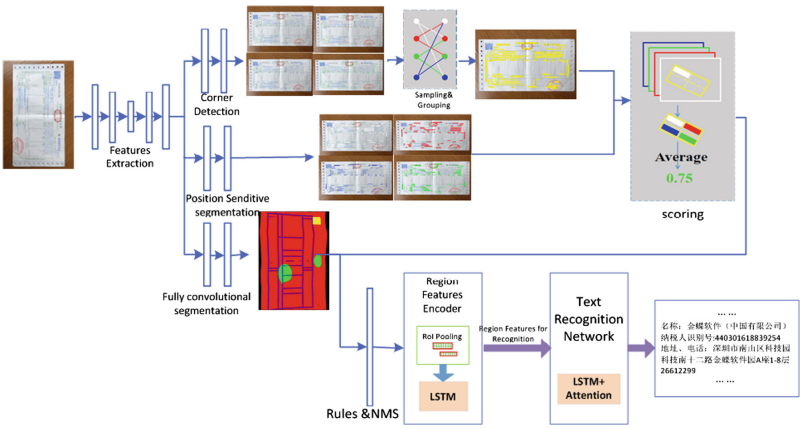

Fig. 1. Model overview. The network takes an image as input, and outputs both text bounding boxes and text labels.

We validate the effectiveness of our method on our accumulated VAT invoice image datasets in the enterprise financial management scenario. The results show the advantages of the proposed algorithm in accuracy and applicability.

The contributions of this paper are three-fold: (1) We propose a unified framework for processing and recognizing the VAT invoices, which can be trained and evaluated end-to-end. (2) Our method can simultaneously handle the challenges (such as rotation, varying aspect ratios, very close instances) in multi-oriented text in VAT invoice images. (3) We take invoice layout information into consideration and use some rule to regress and constrain the text bounding boxes.

2 Related Work

An automatic VAT invoice recognition and processing system essentially includes two tasks: text detection and word recognition. In this section, we present a brief introduction to related works on text detection, word recognition, and text spotting systems for VAT invoice that combine both. The text detection algorithm has developed rapidly in recent years. It can be roughly classified into two categories: horizontal text detection and skew text detection. For horizontal text detection, a number of approaches are proposed to detect words directly in the images using DNN based techniques, and it is similar to the method of object detection. Tian et al. [2] develop a vertical anchor

mechanism, and propose a Connectionist Text Proposal Network (CTPN) to accurately localize text lines in image at ECCV 2016. The latest approach to skew text detection is SegLink [3] and Corner Localization and Region Segmentation proposed by Lyu [4]. SegLink [3] predicts text segments and the linkage of them in a SSD style network and links the segments to text boxes, in order to handle long oriented text in natural scene. Lyu et al. [4] propose to detect scene text by localizing corner points of text bounding boxes and segmenting text regions in relative positions. Word recognition has not made much progress in the last two years. There are two main methods, one of the methods is proposed by Shi et al. [5]. It is a novel neural network architecture, which integrates feature extraction, sequence modeling and transcription into a unified framework, while the another method is presented by Lee et al. [6] which use recursive recurrent neural networks with attention modeling for lexicon-free optical character recognition in natural scene images. Text spotting needs to handle both text detection and word recognition. Li et al. [7] proposed a unified network that simultaneously localizes and recognizes text with a single forward pass, avoiding intermediate processes like image cropping and feature re-calculation, word separation, or character grouping. Combining with specific application scenarios, Xie et al. [1] proposed to use many traditional images processing technology to develop the invoice automatic recognition and processing system.

3 Approach

3.1 Overall Architecture

The whole system architecture is illustrated in Fig. 1. It includes two parts: text detection network (TDN) and text recognition network (TRN). Text detection network aims to localize text in images and generate bounding boxes for words. Text recognition network recognizes words in the detected bounding boxes based on the previous text detection network. Our model is motivated by recent progresses in FPN [8], DSSD [9], Instance FCN models [10] and sequence-to-sequence learning [11, 12], and we also take the special characteristics of text and invoice layout information into consideration. In this section, we present a detailed description of the whole system.

3.2 Text Detection Network

The network of our method is a fully convolutional network (FCN) that plays the roles of feature extraction, corner detection, position-sensitive segmentation and fully convolutional segmentation. Inspired by the good performance achieved by FPN [8] and DSSD [9], we adopt the backbone in FPN/DSSD architecture to extract features. In detail, we convert the fc6 and fc7 in the VGG16 to convolutional layers and name them conv6 and conv7 respectively. Then several extra convolutional layers (conv8, conv9, conv10, conv11) are stacked above conv7 to enlarge the receptive fields of extracted features. After that, a few deconvolution modules proposed in DSSD [9] are used in a top-down pathway (Fig. 2). Particularly, to detect text with different sizes well, we cascade deconvolution modules with 256 channels from conv11 to conv3 (the features

from conv10, conv9, conv8, conv7, conv4, conv3 are reused), and 6 deconvolution modules are built in total. Including the features of conv11, we name those output features F3, F4, F7, F8, F9, F10 and F11 for convenience. In the end, the feature extracted by conv11 and deconvolution modules, which have richer feature representations, are used to detect corner points and predict position-sensitive maps. A large number of candidate bounding boxes can be generated after sampling and grouping corner points. Inspired by [4], we adopt the methods which score the candidate boxes by Rotated Position-Sensitive Average ROI Pooling and detect the arbitrary-oriented text by using position-sensitive segmentation maps.

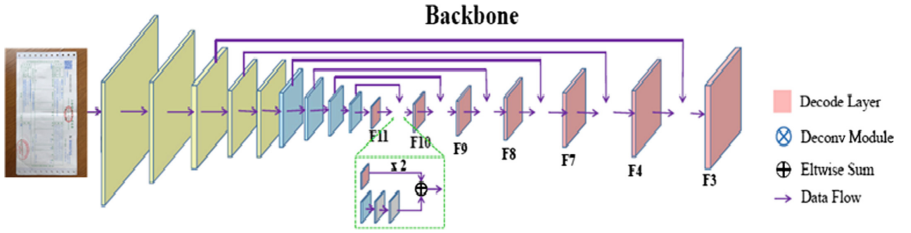

Fig. 2. Network architecture. The backbone is adapted from DSSD [9].

But unlike the above methods [4] that regress text boxes or segments directly, we still added the supplementary method, which uses the invoice layout information in the image (such as the form line, red chop and two-dimensional code.) detected by FCN architecture [13] to constrain the detected bounding boxes and to improve the accuracy and efficiency for text detection. Combine with the above method, we use the NMS and some rules to filter out the candidate boxes with low score and get the RoIs. The detected bounding boxes are merged via NMS according to their textness scores and fed into Text Recognition Network (TRN) for text recognition.

3.3 Text Recognition Network

To process RoIs of different scales and aspect ratios in a unified way, most existing works re-sample regions into fixed-size feature maps via pooling [14]. However, for text, this approach may lead to significant distortion due to the large variation of word lengths. For example, it may be unreasonable to encode short words like “Dr” and long words like “congratulations” into feature maps of the same size. In this work, we propose to re-sample regions according to their respective aspect ratios, and then use RNNs to encode the resulting feature maps of different lengths into fixed length vectors. The whole region feature encoding process is illustrated in Fig. 3.

For an RoI of size h × w, we perform spatial max-pooling with a resulting size of

$$H \times \min(\mathrm{W}_{\max}, 2\mathrm{Hw}/\mathrm{h}), \tag{1}$$

where the expected height H is fixed and the width is adjusted to keep the aspect ratio as 2w/h (twice the original aspect ratio) unless it exceeds the maximum length W_{max}.

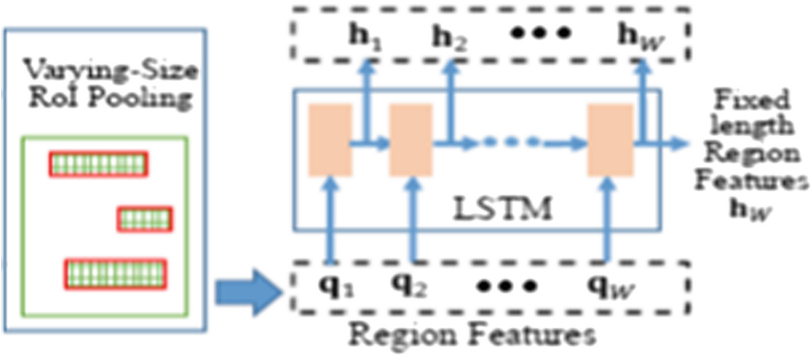

Fig. 3. Region Features Encoder (RFE). The region features after RoI pooling are not required to be of the same size. In contrast, they are calculated according to aspect ratio of each bounding box, with height normalized. LSTM is then employed to encode different length region features into the same size.

Note that here we employ a pooling window with an aspect ratio of 1:2, which benefits the recognition of narrow shaped characters, like 'i', 'l', etc., as stated in [5].

Next, the resampled feature maps are considered as a sequence and fed into RNNs for encoding. Here we use Long-Short Term Memory (LSTM) [11] instead of vanilla RNN to overcome the shortcoming of gradient vanishing or exploding. The feature maps after the above varying-size RoI pooling are denoted as $\mathbf{Q} \in \mathrm{R}^{C\times H\times W}$, where $\mathrm{W} = \min(\mathrm{W}_{max}, 2\mathrm{H}w/\mathrm{h})$ is the number of columns and C is the channel size. We flatten the features in each column, and obtain a sequence $\mathbf{q}_1, \ldots, \mathbf{q}_w \in \boldsymbol{R}^{C\times H}$ which are fed into LSTMs one by one. Each time LSTM units receive one column of feature $\mathbf{q}_t$, and update their hidden state $\mathbf{h}_t$ by a non-linear function: $\mathbf{h}_t = f(\mathbf{q}_t, \mathbf{h}_{t-1})$. In this recurrent fashion, the final hidden state $\mathbf{h}_W$ (with size R = 1024) captures the holistic information of Q and is used as a RoI representation with fixed dimension.

Text recognition aims to predict the text in the detected bounding boxes based on the extracted region features. As shown in Fig. 4, we adopt LSTMs with attention mechanism [12, 15] to decode the sequential features into words.

Firstly, hidden states at all steps $\mathbf{h}_1, \ldots, \mathbf{h}_W$ from RFE are fed into an additional layer of LSTM encoder with 1024 units. We record the hidden state at each time step and form a sequence of $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_W] \in \boldsymbol{R}^{R\times W}$. It includes local information at each time step and works as the context for the attention model.

As for decoder LSTMs, the ground-truth word label is adopted as input during training. It can be regarded as a sequence of tokens $\mathrm{s} = \{\mathbf{s}0, \mathbf{s}1, \ldots, \mathbf{s}_{T+1}\}$ where s_0 and s_{T+1} represent the special tokens START and END respectively. We feed decoder LSTMs with T + 2 vectors: $x_0, x_1, \ldots, x_{T+1}$, where $x_0 = [\mathbf{v}_W; \mathrm{Atten}(\mathbf{V}, 0)]$ is the concatenation of the encoder's last hidden state v_W and the attention output with guidance equals to zero; and $x_i = [\psi(s_{i-1}; Atten(\boldsymbol{V}, \boldsymbol{h}'_{i-1})]$, for $i = 1, \ldots, T+1$, is made up of the embedding $\psi()$ of the (i − 1)-th token s_{i-1} and the attention output guided by the hidden state of decoder LSTMs in the previous time-step $\boldsymbol{h}'_{i-1}$. The embedding function $\psi()$ is defined as a linear layer followed by a tanh non-linearity.

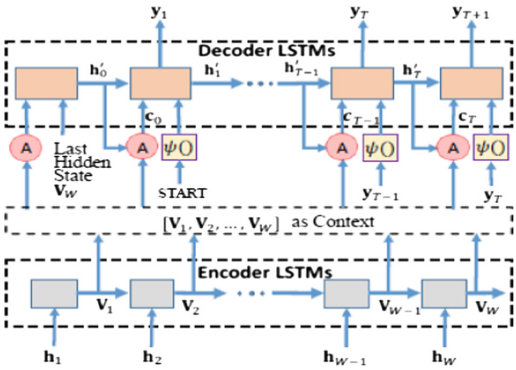

Fig. 4. Text Recognition Network (TRN). The region features are encoded by one layer of LSTMs, and then decoded in an attention based sequence to sequence manner. Hidden states of encoder at all time steps are reserved and used as context for attention model.

The attention function $\mathbf{c}_i = \text{Atten}(\mathbf{V}, \mathbf{h}_i')$ is defined as follows:

$$\begin{cases} g_j = \tanh\left(W_v v_j + W_h h_i'\right), j = 1, \ldots, W, \\ \alpha = softmax\left(w_g^T \bullet [g_1, g_2 \ldots, g_w]\right), \\ c_i = \sum_{j=1}^{W} \alpha_j v_j \end{cases} \tag{2}$$

where $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_W]$ is the variable-length sequence of features to be attended, $\boldsymbol{h}_i'$ is the guidance vector, $\mathbf{W}_v$ and $\mathbf{W}_h$ are linear embedding weights to be learned, α is the attention weights of size W, and $\mathbf{c}_i$ is a weighted sum of input features.

At each time-step $t = 0, 1, \ldots, T+1$, the decoder LSTMs compute their hidden state $\boldsymbol{h}_t'$ and output vector $\boldsymbol{y}_t$ as follows:

$$\begin{cases} \boldsymbol{h}_t' = f(x_t, \boldsymbol{h}_{t-1}') \\ \boldsymbol{y}_t = \varphi\left(\boldsymbol{h}_t'\right) = softmax(W_o \boldsymbol{h}_t') \end{cases} \tag{3}$$

Where the LSTM [11] is used for the recurrence formula f(), and $\mathbf{W}o$ linearly transforms hidden states to the output space, including 26 case-insensitive characters, 10 digits, common standard Chinese characters, a token representing all punctuations like "!" and "?", and a special END token.

At test time, the token with the highest probability in previous output $\boldsymbol{y}_t$ is selected as the input token at step t + 1, instead of the ground-truth tokens $s_1, \ldots, s_T$.

The process is started with the START token, and repeated until we get the special END token.

3.4 Loss Functions and Training

As we demonstrate above, our system takes as input of an image, word bounding boxes and their labels during training. For the final outputs of the whole system, we apply a multi-task loss for both detection and recognition.

$$\mathrm{L} = L_D + L_R \tag{4}$$

Our text detect network model is trained by the corner detection and position-sensitive segmentation simultaneously. The loss function is defined as:

$$L_D = \frac{1}{N_c} L_{conf} + \frac{\lambda_1}{N_c} L_{loc} + \frac{\lambda_2}{N_s} L_{seg} \tag{5}$$

Where $\mathrm{L_{conf}}$ and $\mathrm{L_{loc}}$ are the loss functions of the score branch for predicting confidence score and the offset branch for localization in the module of corner point detection. $\mathrm{L_{seg}}$ is the loss function of position-sensitive segmentation. $\mathrm{N_c}$ is the number of positive default boxes, $\mathrm{N_s}$ is the number of pixels in segmentation maps. $\mathrm{N_c}$ and $\mathrm{N_s}$ are used to normalize the losses of corner point detection and segmentation. λ_1 and λ_2 are the balancing factors of the three tasks. In default, we set the λ_1 to 1 and λ_2 to 10.

We follow the strategy of text recognition which proposed by Lyu et al. [4] and the loss for training text recognition is.

$$\mathrm{L}_R = \frac{1}{\mathrm{N}_c} \sum\nolimits_{i=1}^{N_c} L_{rec}(Y^{(i)}, s^{(i)}) \tag{6}$$

Where s(i) is the ground-truth tokens for sample i and $\mathrm{Y}_{(i)} = \left\{ \mathrm{y}_0^{(i)}, \mathrm{y}_1^{(i)}, \ldots, \mathrm{y}_{T+1}^{(i)} \right\}$ is the corresponding output sequence of decoder LSTMs. $\mathrm{Lrec}(\mathrm{Y}, \mathrm{s}) = -\sum_{t=1}^{T+1} log \boldsymbol{y}_t(s_t)$ denotes the cross entropy loss on $\mathrm{y}_1, \ldots, \mathrm{y}_{\mathrm{T}+1}$, where $\mathbf{y}_\mathrm{t}(\boldsymbol{s}_\mathrm{t})$ represents the predicted probability of the output being $\boldsymbol{s}_\mathrm{t}$ at time step t and the loss on y_0 is ignored.

4 Experiments

In this section, we perform experiments to verify the effectiveness of the proposed method. We use the accumulated VAT invoice image datasets in the enterprise financial management scenario to evaluate the proposed method.

Our method is implemented by using TensorFlow r1.4.1. All the experiments are carried out on a workstation with an Intel Xeon 8-core CPU (2.10 GHz), 2 GeForce GTX 1080 Graphics Cards, and 64 GB RAM. Running on 1 GPUs in parallel, training a batch takes about 1 s. The whole training process takes less than a day.

For different application scenarios of the invoice, scanned invoices and photo invoices achieves different F-measures. The photo invoices is easily influenced by some factors such as size, noise, blur, illumination, contrast and shelter. One contribution of this work is added to the supplementary method,which uses the invoice

layout information in the image to improve the accuracy and efficiency of text detection. To validate its effectiveness, we compare the performance of models "Ours FCN-biLATM+NoLayout" and "Ours FCN-biLATM+Layout". Experiment shows that the model with constrained layout rule significantly better than unconstrained layout rule. As illustrated in Tables 1 and 2, adopting constrained layout rule ("Ours FCN-biLATM +Layout") instead of unconstrained layout rule ("Ours FCN-biLATM+NoLayout") makes F-measures increase around 4%.

Table 1. Results on the scanned invoice image datasets. Precision (P) and Recall (R) at maximum F-measure (F) are reported in percentage.

Method	Precision	Recall	F-measure
Ours FCN-biLATM+NoLayout	89.0	83.0	66.0
Ours FCN-biLATM+Layout	93.3	79.4	85.8

Table 2. Results on the photo invoice image datasets. Precision (P) and Recall (R) at maximum F-measure (F) are reported in percentage.

Method	Precision	Recall	F-measure
Ours FCN-biLATM+NoLayout	66.0	44.7	53.3
Ours FCN-biLATM+Layout	70.8	43.0	53.6

5 Conclusion

In this paper, we have presented an automatic value-added tax (VAT) invoice recognition and processing system. In this system, VAT invoice can be detected and recognized in a single forward pass efficiently and accurately. Experimental results illustrate that the proposed method can produce an impressive performance in the actual projects of enterprises, and the model with constrained layout rule scenarios significantly better than unconstrained layout rule scenarios. One of potential future work is on maintaining images with other bills and documents.

Acknowledgement. This work is partially supported by the technical projects No. c1533411 500138 and No. 2017YFB0802700.

References

1. Xie, Z.G.: Researches on unsupervised image processing of VAT invoices, (Master Thesis) Shanghai Jiao Tong University, Shanghai, China (2015)
2. Tian, Z., Huang, W., He, T., He, P., Qiao, Yu.: Detecting text in natural image with connectionist text proposal network. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9912, pp. 56–72. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46484-8_4
3. Shi, B., Bai, X., Belongie, S.: Detecting oriented text in natural images by linking segments. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 2017

4. Lyu, P., Yao, C., Wu, W., et al.: Multi-oriented scene text detection via corner localization and region segmentation. Journal (2018)
5. Shi, B., Bai, X., Yao, C.: An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. CoRR, abs/1507.05717 (2015)
6. Lee, C.Y., Osindero, S.: Recursive recurrent nets with attention modeling for OCR in the wild. In: Computer Vision and Pattern Recognition, pp. 2231–2239. IEEE (2016)
7. Li, H., Wang, P., Shen, C.: Towards end-to-end text spotting with convolutional recurrent neural networks. Journal (2017)
8. Lin, T.Y., Dollar, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 2017
9. Fu, C.Y., Liu, W., Ranga, A., Tyagi, A., Berg, A.C.: DSSD: Deconvolutional single shot detector. arXiv preprint arXiv:1701.06659 (2017)
10. Dai, J., He, K., Li, Y., Ren, S., Sun, J.: Instance-sensitive fully convolutional networks. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9910, pp. 534–549. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46466-4_32
11. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
12. Shi, B., Wang, X., Lv, P., Yao, C., Bai, X.: Robust scene text recognition with automatic rectification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2016)
13. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3431–3440 (2015)
14. Girshick, R.: Fast R-CNN. In: Proceedings of the IEEE Conference on Computer Vision (2015)
15. Luong, M.T., Pham, H., Manning, C.D.: Effective approaches to attention-based neural machine translation. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (2015)